

optionale Zusatzaufgabe Z 2

Gruppe	
Name(n)	Matrikelnummer(n)

Z 2 Entwicklung eines GPS-Datenloggers

Im abschließenden Aufgabenblatt möchten wir Ihnen die Möglichkeit geben, ein mobiles eingebettetes System zu entwickeln: einen **GPS-Datenlogger**. Zu diesem Zweck steht Ihnen das **GlobalTop PA6H** GPS Empfangsmodul zur Verfügung. Das GPS Modul basiert auf dem **MTK3339** Chipsatz. Informationen zum Funktionsumfang und zum Kommunikationsprotokoll finden Sie im **Datenblatt**.

In **Abbildung 1** sind die Anschlusspins der Platine des GPS Moduls dargestellt. Für die Verwendung des Moduls wird nur ein Teil davon benötigt. Führen Sie die notwendige Verdrahtung auf dem Steckbrett in Anlehnung an **Abb. 2** durch.

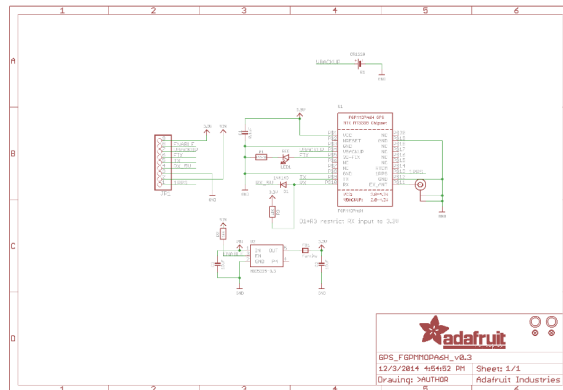
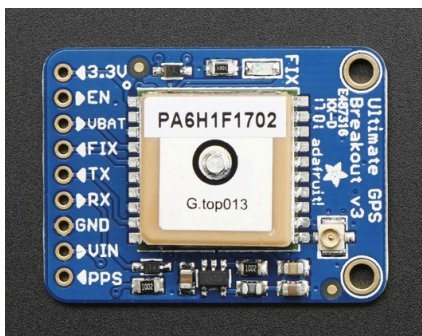


Abbildung 1: Anschlussbelegung des PA6H/MTK3339 Breakout-Boards.

Die vom GPS Modul über die serielle Schnittstelle ausgegebenen Daten entsprechen einer Teilmenge des **NMEA-0183**-Protokolls. Informationen zu den verwendeten Datensätzen dieses Protokolls finden Sie im **Datenblatt** des GPS Moduls. Weitere Informationen zum NMEA-0183-Protokoll finden Sie unter de.wikipedia.org/wiki/NMEA_0183 oder auch im **NMEA Reference Manual** der *SiRF Technology*.

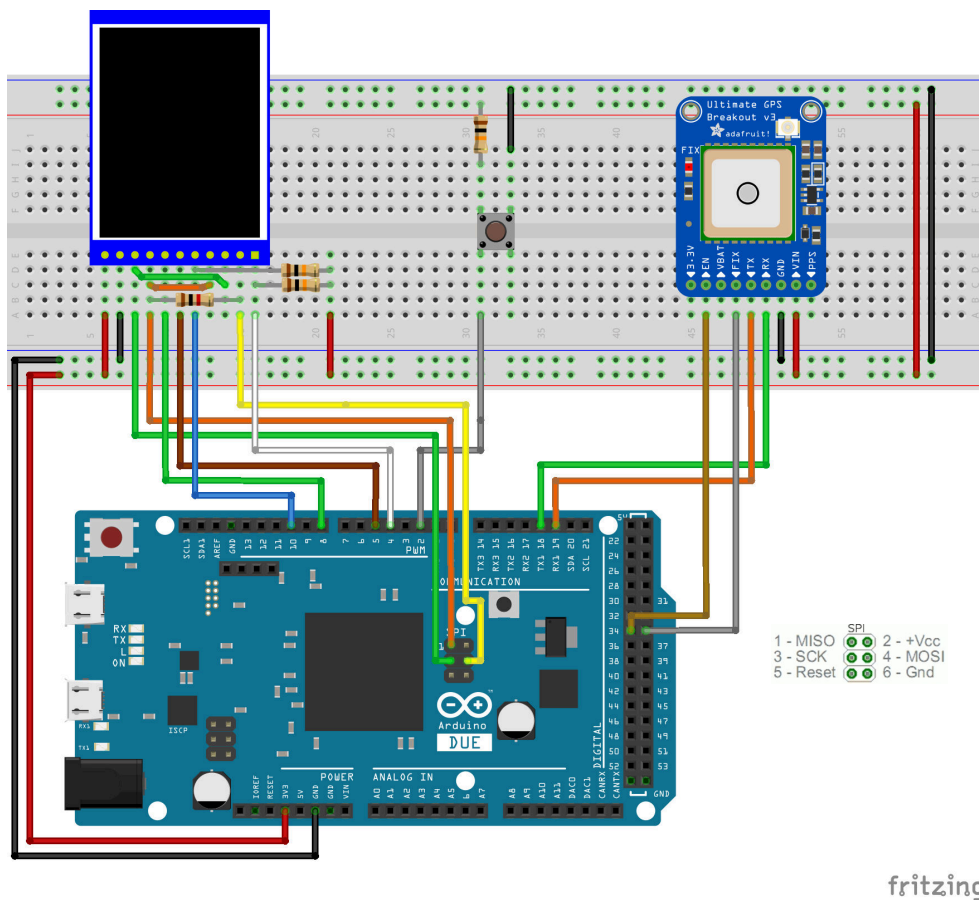


Abbildung 2: Vorschlag für die Verdrahtung des Versuchsaufbaus.

Für die Lösung der Aufgabe wird empfohlen, den Versuchsaufbau gemäß Abbildung 2 zu verdrahten. Schließen Sie das TFT-Display, den Taster nach bekanntem Schema und das GPS Modul an das Arduino Board an. Beachten Sie bei der Verdrahtung des GPS Moduls folgende Hinweise:

- Verbinden Sie VIN mit **3,3 V** und stellen Sie zusätzlich die Masseverbindung (GND) her.
- Verbinden Sie RX und TX **gekreuzt** mit den Anschlüssen einer beliebigen seriellen UART Schnittstelle des Prozessors des Arduino-Boards (warum gekreuzt?). Empfohlen wird die zweite serielle UART Schnittstelle (RX1/TX1).
- der Enable-Pin EN ist optional. Er wird seitens des Breakout-Boards über einen 10k-PullUp-Widerstand auf VCC gezogen. Wird der Pin auf GND gelegt, wird das GPS deaktiviert.
- die Verwendung des FIX-Pins ist ebenfalls optional. Hier liegt das Signal an, welches die auf dem Breakout-Baord befindliche Fix-LED treibt (siehe Abb. 1).

Aufgabe Z 2.1

Entwickeln Sie ein Programm, welches die Parameter des vom GPS Modul gelieferten NMEA-0183 Datensatzes **GPGGA** sinnvoll auf der Ausgabe des seriellen Monitors der Arduino IDE präsentiert.

Wichtig: Implementieren Sie zunächst eine Routine zur Berechnung der erwarteten Prüfsumme, damit eine Verifikation des empfangenen Datensatzes möglich wird und Ihr Parser nur gültige Daten verarbeitet. Extrahieren Sie daraufhin die notwendige Information zur Bestimmung von:

- **Breiten- und Längengrad** der Position.
- Anzahl der für die Positionsbestimmung verwendeten (d.h. sichtbaren) Satelliten.

Implementieren Sie zusätzlich die Behandlung des Nichtvorhandenseins von GPS-Fix Information (d.h. fehlender Fix, mangels sichtbarer Satelliten). **Wichtig:** Positionieren Sie den Aufbau zum Testen Ihres Programms am Fenster, da das GPS Modul selbstverständlich nur hier die Möglichkeit hat, Daten mehrerer Satelliten zu empfangen. Wieviele Satelliten werden mindestens benötigt, um eine Position zu bestimmen?

Aufgabe Z 2.2

Erweitern Sie Ihr Programm um Funktionalität zum Abspeichern von Daten auf der microSD Speicherkarte. Verwenden Sie die bereits bekannte SD Bibliothek. Machen Sie sich noch einmal mit dem Funktionsumfang der Bibliothek vertraut (www.arduino.cc/en/Reference/SD). Schauen Sie sich insbesondere die folgenden, zum Schreiben von Daten benötigten, Funktionen an:

* SD.begin(<sd-pin>)	→ SD.begin
* SD.open(<file path>, <mode>)	→ SD.open
* <file>.write(<data>)	→ file.write
* <file>.close()	→ file.close

Implementieren Sie ein Verfahren zur Aufzeichnung der extrahierten Positionsdaten in einer Datei auf der microSD Speicherkarte. Benutzen Sie die Betätigung des Tasters als Ereignis für den Start/Stopp der Aufzeichnung. Vergessen Sie dabei bitte nicht zu überprüfen, ob ein GPS-Fix vorliegt. Schreiben Sie die Breiten-/Längengrad-Paare (**durch ein Komma getrennt, zeilenweise**) in eine von Ihnen benannte Datei auf der microSD Speicherkarte:

```
52.027155,11.283824-
50.519239,13.615708-
49.204180,16.442907-
...
```

Sollten Sie anstelle des in Aufg. 2.3 vorgeschlagenen „GPSvisualizer“ einen unter Android verfügbaren GPS-Viewer verwenden wollen, der NMEA-Datensätze verarbeiten kann, schreiben Sie anstelle der Breiten-/Längengrad-Paare den entsprechenden bereits validierten NMEA-Datensatz **GPGGA** zeilenweise in die Log-Datei.

Beachten Sie: Die Arduino SD Bibliothek schränkt die Benennung der Dateien durch das **8.3 Format** ein (de.wikipedia.org/wiki/8.3).

Sollte der GPS-Fix während der Aufzeichnung verloren gehen, terminieren Sie das Logging. Überlegen Sie sich eine Vorgehensweise zur Wiederaufnahme der Aufzeichnung, z.B. Anhängen neuer Daten an die vorhandene Datei oder Erstellen einer neuen Datei mit einem Zähler im Namen (z.B. `gps001.txt`).

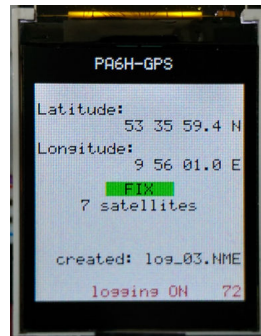


Abbildung 3: Möglichkeit zur Präsentation der Daten auf dem TFT-Display.

Verwenden Sie das TFT-Display zur Ausgabe der extrahierten Daten sowie des Zustands der Aufzeichnung. Abbildung 3 zeigt eine Möglichkeit zur Präsentation dieser Daten.

Machen Sie Ihren GPS-Datenlogger mobil, indem Sie das Arduino-Board beispielsweise an eine Powerbank anschließen. Begeben Sie sich auf das Campusgelände zum Testen des von Ihnen entworfenen GPS-Datenloggers und zeichnen Sie eine Route auf.

Aufgabe Z 2.3

Um die gespeicherten Positionsdaten des Datenloggers auslesen zu können, verwenden Sie Ihre zu Aufgabenblatt 6 entwickelte Routine `outputFileToSerial()` zur Ausgabe einer Datei auf dem seriellen Monitor der Arduino IDE. Für die Ausgabe können Sie auch den *Native USB-Port* des DUE verwenden. Speichern Sie den ausgegebenen Inhalt in einer Textdatei auf Ihrem Rechner. Rufen Sie die Webseite www.gpsvisualizer.com in einem Web-Browser auf und laden Sie die erstellte Datei zur Visualisierung in Google Maps hoch.

Sollten Sie einen GPS-Viewer auf Ihrem Smartphone installiert haben, der das oben beschriebene spaltenorientierte Textformat oder NMEA-Datensätze verarbeiten kann, können sie auch gerne diesen benutzen.